Ogrodje Metasploit

Franci Šacer

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova 17, 2000 Maribor

franci.sacer@student.um.si

Povzetek – V današnjih časih informacijska varnost postaja vse bolj pomembna. Vse več zaupnih informacij npr. kot so zdravstvena poročila in ostali pomembnejši dokumenti se shranjuje elektronsko na računalnikih, ki so pogosto priključeni tudi v internet. Zagotavljanje varnosti sistemov, ki so prepleteni tako s staro kot novejšo tehnologijo, ni preprosto. Eden izmed pomembnejših dejavnikov, ki pripomorejo k zagotavljanju varnosti, je penetracijsko testiranje. Ker delo testerjev ni preprosto si pri tem lahko pomagajo z raznimi orodji. Eno izmed takšnih orodij je ogrodje Metasploit. V tej seminarski nalogi predstavim nekatere osnovne podatke o njem, zgodovino razvoja, njegovo modularno arhitekturo, modul Meterpreter in grafični vmesnik Armitage.

Abstract – Nowadays, information security is becoming more and more important. Increasing amount of confidential information, like medical reports and other important documents, is being stored electronically on computer systems and those systems are often connected to the Internet. Ensuring security in systems intermingled with old and new technology isn't simple. One of the more important factors for ensuring security is penetration testing. Because the work of penetration testers isn't trivial, they can help themselves by using several different tools. One of those tools is the Metasploit framework. In this seminar paper, I present some basic information about the framework, history of development, it's modular architecture, the Meterpreter module and the Armitage GUI.

1 Uvod

Informacijske tehnologije so zapleteno področje, prepleteno z zastarelo in novo tehnologijo, sistemi, programsko opremo in protokoli. Varovanje današnjih komercialnih sistemov ni le posodabljanje, požarni zidovi in izobraževanje uporabnikov, ampak zahteva pogosto validacijo tega kar deluje in kaj ne. To pokriva penetracijsko testiranje. (O'Gorman, Kearns, & Aharoni, 2011)

Penetracijsko testiranje je eden izmed glavnih dejavnikov omrežne varnosti danes. Vključuje celostno analizo sistema z realnimi testi varnosti. Ta pomaga pri prepoznavanju potencialnih ranljivosti sistema oz. njegovih sestavnih delov (tj. tako programska in strojna oprema). Razlog, zakaj je penetracijsko testiranje pomembno lahko najdemo v tem, da nam pomaga prepoznati ranljivosti iz hekerjevega pogleda. (Singh, 2012)

Poznamo več oblik penetracijskih testov na podlagi črne in bele skrinje (razlika je v tem koliko podatkov nam je dano o tarči) ali gre za zunanji penetracijski test (lahko gre tudi za zunanji test na podlagi črne skrinje) ali pa za test znotraj omrežja (interni). (Singh, 2012)

Ker pa je penetracijsko testiranje zahtevno delo (brez avtomatizacije bi to zgolj temeljilo na poskusih in napakah) in zahteva razmišljanje na nivoju kriminalca so se razvila razna orodja, z namenom, da pomagajo penetracijskim testerjem pri svojem delu. (O'Gorman, Kearns, & Aharoni, 2011) Ta orodja avtomatizirajo vsaj nekatere dele penetracijskega testiranja. Eno izmed takšnih orodji je odprtokodno ogrodje Metasploit, ki poskuša avtomatizirati večino

korakov takšnega testiranja (skeniranje za odkrivanje ranljivih sistemov, module za izkoriščanje ranljivosti in izvajanje kode po izkoriščeni ranljivosti, orodja za izogibanje zaznavanju vdorov). Glavna prednost ogrodja je v modularni arhitekturi in v načinu kako je bil razvit. (Maynor, 2011) Obstajajo tudi komercialne različice tega ogrodja, ki dodajo spletni vmesnik in še nekaj naprednih funkcionalnosti, a teh nisem podrobneje opisoval. (Rapid7, 2017)

V naslednjem poglavju sem na splošno opisal ogrodje. Tretje poglavje opisuje zgodovino razvoja ogrodja. Četrto poglavje opisuje arhitekturo ogrodja. V petem poglavju je opisan eden izmed najpopularnejših modulov - Meterpreter, ki se uporablja za izvajanje kode po izkoriščanju ranljivosti (angl. payload). V zadnjem poglavju sem pregledal grafični vmesnik Armitage za Metasploit.

2 Splošno o ogrodju

Ogrodje Metasploit (angl. Metasploit Framework) je odprtokodna platforma, za razvoj in izvajanje programov za izkoriščanje ranljivosti (angl. exploit) in pa programov, ki se izvedejo ob izkoriščeni ranljivosti (angl. payload). Poganjamo ga lahko tako na Windowsu, kot na Unix sistemih. Ima tudi dodatne funkcionalnosti kot so skeniranje za identifikacijo ranljivih sistemov, izogibanje sistemom za zaznavanje in preprečevanje vdorov, generiranje poročil itd. Poleg orodja pa zraven dobimo še več tisoče modulov, ki so namenjeni izkoriščanju ranljivosti. Ima eno izmed največjih zbirk testiranih modulov za izkoriščanje ranljivosti in dobi več milijonov prenosov na leto. Ogrodje je del projekta Metasploit in je razvito v programskem jeziku Ruby, ter je tudi eden izmed najkompleksnejših projektov, ki so bili do danes izgrajeni na osnovi tega programskega jezika. (Singh, 2012) Odložišče kode pa lahko najdemo na GitHub-u z URL naslovom: https://github.com/rapid7/metasploit-framework. Trenutno ga vzdržuje podjetje Rapid7. (Rapid7, 2012)

Osnovni delovni tok pri uporabi ogrodja skoraj, da sledi standardu PTES (angl. Penetration Testing Execution Standard). Koraki delovnega toka so:

1. Skeniranje omrežja in skeniranje storitev na tarči (opcijsko)
2. Izbira modula za izkoriščanje ranljivosti (angl. exploit)
3. Obkljukamo za preverjanje, da je sistem res ranljiv (opcijsko)
4. Izbira modula, ki se izvede ob izkoriščeni ranljivosti (angl. payload) - npr. povratna lupina ali VNC strežnik
5. Izbiranje tehnike kodiranja, ki omogoči, da sistem za preprečevanje vdora ignorira »payload«
6. Izvajanje modulov

(Skupina PTES, 2009) (Wikipedia, 2017) (Rapid7, 2012)

3 Zgodovina razvoja

Metasploit je v osnovi zasnoval HD Moore, ki ga je zaposlilo neko ameriško podjetje za varnost. Ko je Moore spoznal, da je porabljal veliko njegovega časa pri validaciji in prilagojevanjem javnih programov za izkoriščanje ranljivosti, je začel z razvojem fleksibilnega ogrodja, ki ga je enostavno vzdrževati. Namen je bil ustvarjanje in razvoj programov za izkoriščanje ranljivosti. Izdal je njegovo prvo različico (bazirana na Perl-u) oktobra 2003, ki je vsebovala 11 modulov. (O'Gorman, Kearns, & Aharoni, 2011)

S pomočjo Spoonm-a, je Moore izdal ponovno napisan projekt Metasploit 2.0 aprila 2004. Ta različica je vsebovala 19 »exploitov« in več kot 27 »payloadov«. Kmalu po izdaji se je Matt Miller pridružil razvojni ekipi za Metasploit. Ko je projekt začel rasti, je delo na ogrodju

Metasploit dobilo močno podporo iz strani skupnosti za informacijsko varnost in hitro postalo nujno orodje za penetracijsko testiranje ter za izkoriščanje varnostnih ranljivosti. (O'Gorman, Kearns, & Aharoni, 2011)

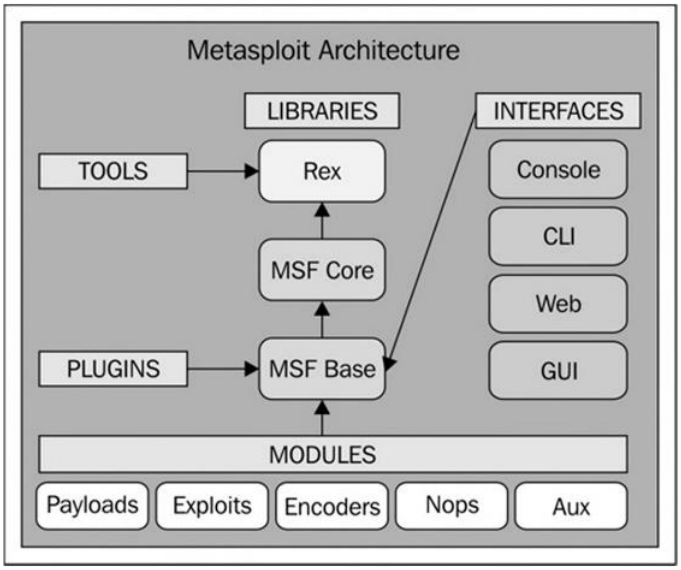
Po še enem kompletnem ponovnem razvoju v Ruby-u, je ekipa Metasploit izdala različico 3.0 v letu 2007. Migracija iz Perla na Ruby je trajala 18 mesecev z 150.000 vrstic kode. Z izdajo 3.0 se je ogrodje po uporabi zelo razširilo v varnostni skupnosti, povečalo se je pa tudi število prispevkov uporabnikov. Ta različica je tudi kot prva vključevala orodja za »fuzz« testiranje (tehnika avtomatskega testiranja programske opreme, ki vključuje vnašanje neveljavnih, nepričakovanih ali naključnih vrednosti kot vhod). (Wikipedia, 2017) (O'Gorman, Kearns, & Aharoni, 2011)

V jeseni 2009 je Metasploit prevzelo vodilno podjetje v skeniranju ranljivosti Rapid7, kar je Mooru omogočalo zgraditi ekipo, ki bi se fokusirala samo na razvoj ogrodja Metasploit. Od prevzema se je ogrodje razvijalo bolj hitro kot bi si lahko kdorkoli predstavljal. Podjetje Rapid7 je izdalo tudi 2 komercialna produkta, ki bazirata na ogrodju Metasploit: Metasploit Express in Metasploit Pro. Metasploit Express je vitkejša različica z grafičnim vmesnikom in dodatno funkcionalnostjo kot je generiranje sporočil med drugimi uporabnimi funkcionalnostmi. Metasploit Pro je pa razširjena različica Metasploit Express, ki dodaja zmožnosti sodelovanje in skupinskega penetracijskega testiranja ter še mnogo več uporabnih funkcionalnosti kot je vzpostavitev VPN omrežja z enim pritiskom. Izdana je pa bila tudi zastonjska Community različica, namenjena študentom in malim podjetjem, ki ponuja okrnjene funkcionalnosti komercialnih produktov. (O'Gorman, Kearns, & Aharoni, 2011)

Različica 4.0 je bila izdana avgust 2011 z raznimi novimi funkcionalnostmi in moduli. Danes se Metasploit še vedno aktivno razvija z vse večjim naborom modulov za izkoriščanje svežih ranljivosti. (Rapid7, 2011)

4 Arhitektura

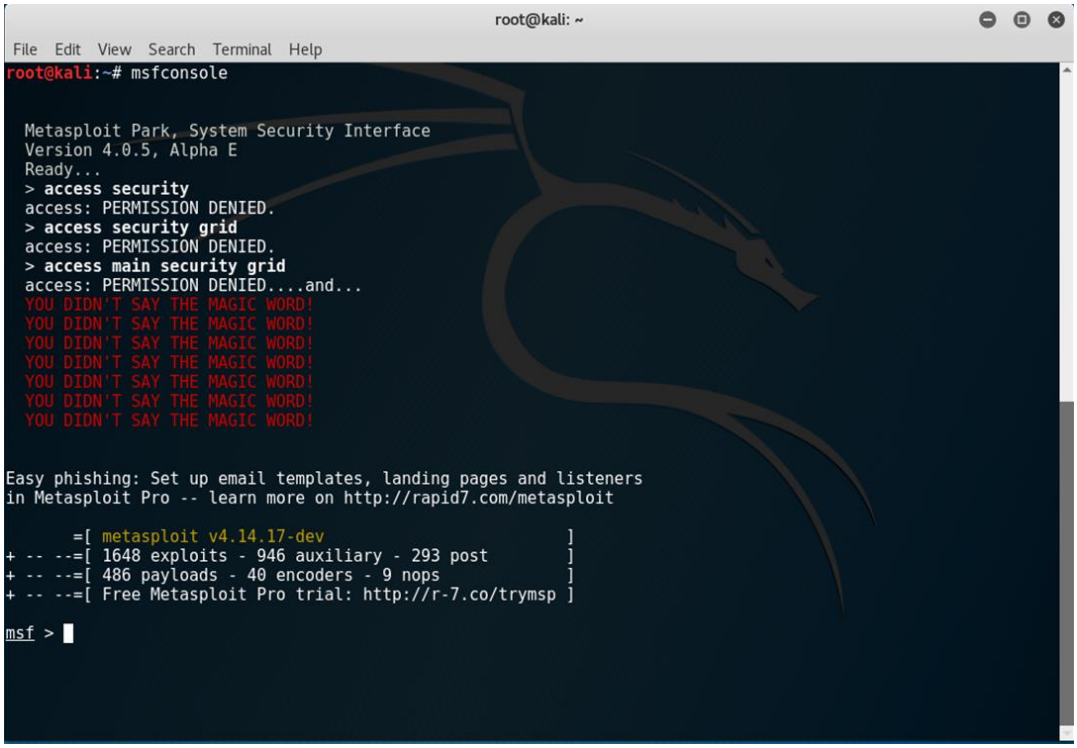
Ogrodje Metasploit ima modularno arhitekturo. Kar pomeni, da so programi za izkoriščanje ranljivosti, programi za izvajanje kode po izkoriščeni ranljivosti, programi za kodiranje, Nop generatorji za skrivanje pred sistemi za preprečevanje vdorov in ostali moduli ločeni od jedra ogrodja. To pomeni, da ogrodje zelo dobro podpira razširljivost. (Singh, 2012)



Slika 1: Arhitektura ogrodja Metasploit

Kot lahko vidimo na sliki 1, ogrodje temelji na nekaj različnih knjižnicah, ki so ključne za delovanje ogrodja. Osnova je knjižnica Ruby Extension (Rex), ki vključuje razrede za delo z omrežnimi vtiči, raznimi protokoli, podsistem za dnevnike in ostale priročne razrede. Knjižnico Rex razširja knjižnica MSF Core, ki definira ogrodje Metasploit in njegove API vmesnike za module, seje in vtičnike. To jedrno knjižnico pa razširja osnovna knjižnica (MSF Base), ki ponuja ovoj za ogrodje, da so API-ji na višjem nivoju abstrakcije in so prijaznejši za uporabo. Ta ponuja npr. module za serializacijo podatkov modulov v druge izhodne formate. (A STUDY OF METASPLOIT TOOL, 2016)

Osnovna knjižnica je razširjena z osnovo za uporabniške vmesnike (UI). Metasploit podpira razne uporabniške vmesniki od konzolnih, grafičnih pa do spletnih vmesnikov. Poleg ogrodja dobimo štiri vmesnike: msfconsole (glavni konzolni vmesnik), msfcli (zastareli konzolni vmesnik), msfweb (spletni vmesnik) ter msfgui (okenski program). Vmesnik msfconsole lahko vidimo na sliki 2. Obstajajo pa še nekateri drugi grafični vmesniki za ogrodje Metasploit. Eden od takšnih je okenski vmesnik Armitage za Metasploit, ki je razvit ločeno od podjetja Rapid7. Ena izmed glavnih prednosti, ki jih ponuja je hkratno izvajanje več sej in deljenje podatkov na eni instanci Metasploita. Tega sem namreč podrobneje predstavil v zadnjem poglavju. (Singh, 2012)



Slika 2: Konzolni vmesnik msfconsole

Moduli za izkoriščanje novih ranljivosti so pa tisto, kar daje ogrodju veliko prednost pred ostalimi podobnimi ogrodji. Ti se namreč razvijajo neprekinjeno. Eden izmed zadnjih razvitih modulov je za izkoriščanje ranljivosti MS17-010 EternalBlue, ki izkorišča ranljivost v SMB protokolu (za skupno rabo datotek), odkrita in izkoriščena iz strani ameriške agencije NSA. Tega je namreč izkoriščala tudi izsiljevalska programska oprema Wannacry ter EternalRocks. (EternalBlue: Metasploit Module for MS17-010, 2017) Module ogrodje razvršča v naslednje podskupine:

- Programi za izkoriščanje ranljivosti (angl. exploit) – glavni fokus ogrodja
- Programi, ki se izvedejo ob izkoriščeni ranljivosti (angl. payload) – Če se program za izkoriščanje ranljivosti uspešno izvede, imamo veliko možnosti, kaj lahko naredimo s

kompromitiranim sistemom. To vključuje od dodajanje dodatnega uporabnika, izvajanje specifičnih ukazov, dostop do ukazne vrstice, postavljanje VNC strežnika pa do uporabe modula Meterpreter, ki je podrobneje predstavljen v petem poglavju.

- NOP generatorji – Pri ranljivosti prekoračitve izravnalnika pogosti ni znana točna lokacija, kjer se bo program izvajal naprej. Zato so programom za izkoriščanje ranljivosti ranljivostim pogosto dodani NOP (Not Operation) ukazi iz zbirnika. Ti tvorijo nekakšno pot do naslova, kjer se začne izvajati dejanski program. Takšne značilnosti pogosto zaznajo sistemi za zaznavanje vdorov. Zato, da se sistem ne sproži, lahko z NOP generatorji zakrijemo te sekvence NOP ukazov.
- Enkoderje – Podobno kot sekvence NOP instrukcij, so lahko tudi programi, ki se izvedejo ob izkoriščeni ranljivost, zaznani s sistemi za zaznavanje vdorov. S kodiranjem teh programov lahko zagotovimo, da njihovi podpisi niso prepoznani in lahko prosto prečkajo omrežje.
- Pomožni moduli – Nekateri moduli pa niso namenjeni izkoriščanju ranljivosti, ampak so v pomoč pri penetracijskem testu. Takšni moduli so za skeniranje omrežja za ranljivosti, pridobivanje seznama storitev na računalniku, pošiljanje nepričakovanih podatkov (»fuzz« testiranje), povezovanje z strežniki in še mnogo drugih opravil. Ideja je avtomatizirati celoten cikel penetracijskega testa in še po možnosti generirati dobro poročilo.

(Maynor, 2011)

Vtičniki so eden izmed novih konceptov ogrodja. V primerjavi z moduli, so vtičniki namenjeni razširjanju ogrodja samega. To še dodatno poveča uporabnost tovrstnega orodja.

Kot primer, vtičnik je lahko razvit z namenom, da doda dodaten ukaz v konzolni vmesnik. Napredni vtičniki pa lahko avtomatizirajo nek celoten postopek. To je stvar kreativnosti raziskovalca varnostnih ranljivosti. Vtičnik bi lahko bil razvit da požene enega ali več izvidnih modulov, poišče računalnike na omrežje, ugotovi izvajajoče storitve. Lahko tudi sprejme vhode in ugotovi vse možne ranljivosti, ki jih lahko izkoristimo na tarči. Potem pa bi lahko po možnosti zagnal raznovrstne programe za izkoriščanje teh ranljivosti s poskušanjem različnih nastavitev. Med vsem tem pa bi lahko shranjeval vse podatke v podatkovno bazo in iz tega generiral lepo poročilo. (Maynor, 2011)

Ogrodje Metasploit podpira razne relacijske podatkovne baze s pomočjo vtičnikov. V njej se hranijo vsi dnevniki povezani z izvedenimi testi. Iz njih lahko lažje generiramo neko smiselno poročilo, ker imamo potem na voljo vse podatke o tem kar smo počeli. Trenutno podpira PostgreSQL in SQLite. Preden poženemo Metasploit je v večini primerov potrebno nastaviti podatkovno bazo in jo inicializirati za uporabo z ogrodjem. Nastavimo lahko tudi delovno okolje, da ločimo posamezne projekte penetracijskega testiranja in lažje upravljamo z večjo količino podatkov. (Maynor, 2011) (Rapid7, 2012)

5 Meterpreter

Pogosto je faza po izkoriščeni ranljivosti pozabljena oziroma ji pogosto ne dajejo takšnega pomena. Ta faza je zahtevnejša kot si mogoče na videz predstavljamo, sploh, če je potrebno sisteme izkoristiti na skriven in fleksibilen način. Nekaj izmed preprek, ki jih lahko pri tem naletimo so:

- Pri poganjanju dodatnega procesa, se bo ta proces pokazal na seznamu izvajajočih procesov upravitelja opravil od operacijskega sistema. Že poskus nalaganja trojanskega

konja bi pustil dovolj sledi izkušenemu forenzičnemu preiskovalcu. Poganjanje ukazne lupine bi tudi v nekaterih sistemih za zaznavanje vdorov sprožilo alarm. Poleg rdečega signala je lahko tudi lupina, ki bi jo pognali zelo omejena (*chroot* okolje z omejenim dostopom do ukazov in knjižnic).

- Torej, če se želimo izogniti protivirusnim rešitvam se je potrebno izogniti ustvarjanju novih procesov in ustvarjanju novih datotek. Oboje pošilja rdeč signal takšnim rešitvam. Torej se mora takšen program izvajati znotraj izkoriščenega procesa, da ga ne bo mogoče zaznati. Pri tem so nam v pomoč tehnike za vrivanje DLL knjižnic v izvajajoč se program.
- Pogosto so akcije in programi, ki se izvedejo ob izkoriščeni ranljivosti že fiksno določeni v kodi programa za izkoriščanje ranljivosti. Torej bi se morali že vnaprej odločiti ali želimo speljati povratno lupino, dodati uporabnika na okužen sistem enostavno izvesti specifičen ukaz, ko izkoriščanje ranljivosti uspe. To ima zelo slabo stran v tem, da ne moremo biti prilagodljivi oziroma dinamični pri izvajanju napada.

(Maynor, 2011) (Holik, Horalek, Marik, Neradova, & Zitta, 2014)

Pri izogibanju vseh teh omejitev nam priskoči na pomoč modul Meterpreter. To je mogoče zaradi načina implementacije modula. Kot prvo se Meterpreter lahko izogne ustvarjanju novih procesov, ker se izvaja v kontekstu okuženega procesa. To doseže s tehniko vrivanja DLL knjižnic. Zaradi tega se lahko izogne tudi *chroot* okolju. V nekaterih primerih se lahko okužena aplikacija tudi izvaja dalje. Prav nam pride tudi možnost pisanja razširitev za omenjeni modul. (Rapid7, 2004)

Meterpreter nam poda API-je, da lahko napadalec sprogramira razne razširitve za izvajanje kode po izkoriščeni ranljivosti v lupini Meterpreter-ja. Lupina modula je v bistvu napadalska platforma, ki je vrinjena v spomin izvajajočega se procesa. API je zgrajen na način, da omogoča minimalno spremembo v stanju sistema, da je čim manj opazen. (Maynor, 2011)

V Meterpreter je vgrajen že kar obsežen nabor ukazov, ki demonstrirajo česa vsega je ta fleksibilen modul zmožen. Ti ukazi nam omogočajo:

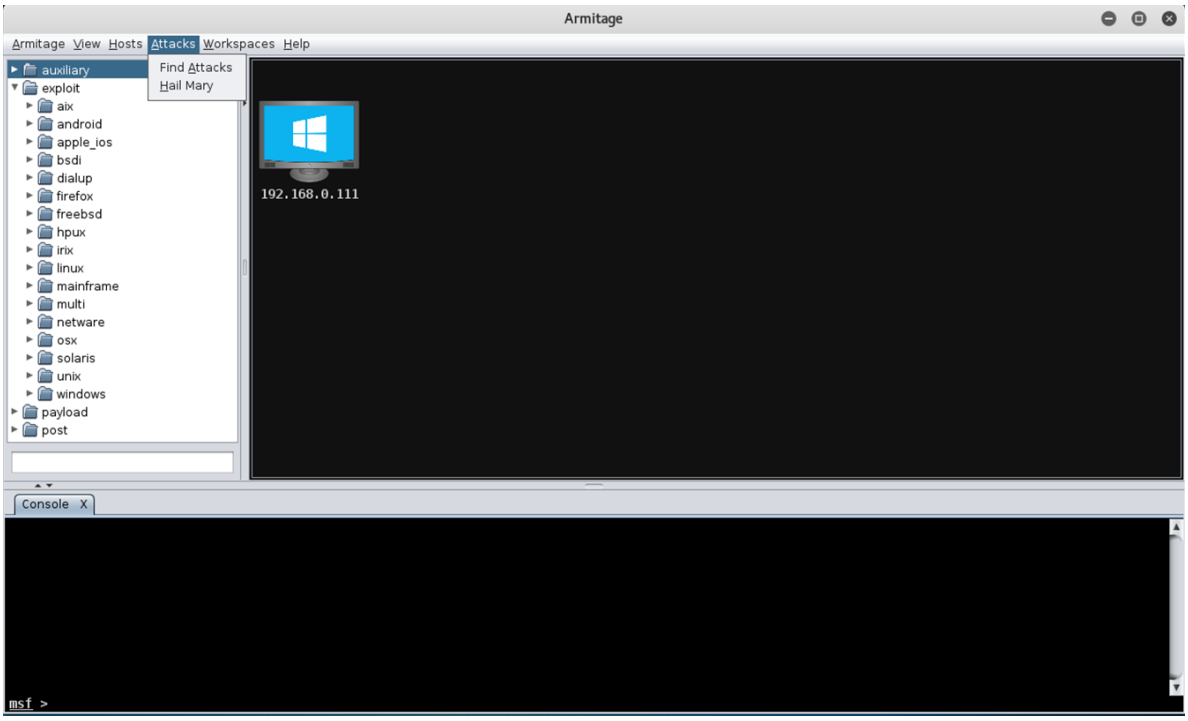
- pridobivanje seznama datotek in map,
- nalaganje in prenašanje datotek,
- urejanje datotek in njihovih atributov,
- pridobivanje zgoščenih vrednosti gesel,
- brisanje dnevnikov dogodkov,
- spreminjanje usmerjevalne tabele,
- izvajanje poljubnih ukazov,
- zajemanje zaslonskih slik in video posnetka iz spletne kamere,
- migriranje modula na različne procese,
- nadziranje miške/tipkovnice,
- pošiljanje WMI (Windows Management Instrumentation) poizvedbe – ta omogoča upravljanje računalnika (zaganjanje procesov takoj ali po razporedu, ponovni zagon ipd.),
- nadziranje odložišča,
- nadzor storitev,
- kraja žetonov,
- ustvarjanje računov,
- in še mnogo več.

Kot lahko vidimo nam Meterpreter omogoča najrazličnejše načine interakcije s tarčo. Lahko upravljamo z datotekami, kar ima lahko različne posledice glede na tarčo. Manipuliramo lahko

tudi s procesi in storitvami. To nam npr. omogoča, da onesposobimo protivirusno zaščito in v sistem namestimo zadnja vrata. Ker lahko manipuliramo z usmerjevalnimi tabelami lahko to pomeni napad mož v sredini. Pobrišemo pa lahko tudi sledi za seboj tako, da nas forenzični preiskovalec težje ujame. Ker lahko ukrademo žetone pa lahko pridobimo dostop do virov (datotek), do katerih predhodno nismo imeli dostopa (v sledeh kot legitimen uporabnik). (Holik, Horalek, Marik, Neradova, & Zitta, 2014)

6 Armitage za Metasploit

Armitage je grafični uporabniški vmesnik za upravljanje s kibernetskimi napadi v okviru ogrodja Metasploit. Razvil ga je Raphael Mudge in je z svojim bogatim grafičnim vmesnikom na voljo zastonj. Ta vmesnik je razviden na sliki 3. Z njim lahko vizualiziramo tarče in priporočene programe za izkoriščanje ranljivosti. Izpostavi nam pa tudi naprednejše funkcionalnosti ogrodja. Armitage je namenjen predvsem tistim, ki ogrodja Metasploit ne uporabljajo vsak dan. Z njim se lahko naučimo tudi naprednejših funkcionalnosti ogrodja. Armitage funkcionalnosti kategorizira enako kot Metasploit. (O'Gorman, Kearns, & Aharoni, 2011) (Darknet, 2010)



Slika 3: Grafični vmesnik Armitage za Metasploit

Za odkrivanje tarč lahko uvozimo rezultate skeniranja uvozimo ali pa ga direktno izvedemo. Uvožene gostitelje nam vizualno predstavi. Pomaga nam tudi pri izkoriščanju ranljivosti tako, da nam predlaga programe za izkoriščanje le-teh. Za te lahko tudi preverimo ali je tarča dejansko ranljiva. Če nam poskušanje ne gre od rok lahko zaženemo funkcijo »Hail Mary«, ki požene celotno bazo ranljivosti proti tarči, katere uporaba je sicer odsvetovana. Armitage ima tudi dobro integracijo z Meterpreter-jem in nam ponudi kontekstni meni za uporabo modula. Armitage pa nam omogoča tudi nastaviti odskočno desko (angl. pivot) – pri tem uporabimo kompromitiranega gostitelja za platformo napada na ostale gostitelje in s tem uspešneje manevriramo po omrežju. Uporaba te funkcionalnosti je mogoča tudi za eksterna orodja. Pri izvajanju več akcij pa nam pomaga z zavihki in delovnimi okolji. (Darknet, 2010)

7 Zaključek

Informacijske tehnologije so zapleteno področje, prepleteno z zastarelo in novo tehnologijo, sistemi, programsko opremo in protokoli. Varovanje takšnih omrežij ni tako preprosto. Penetracijsko testiranje je eden pomembnih faktorjev, ki prispevajo k varnosti. To so realni testi varnosti sistema, ki nam povedo kaj pri varnosti dejansko deluje in kaj ne. Delo penetracijskih testerjev pa ni vedno lahko, zato obstajajo ogrodja, ki poskušajo vsaj delno avtomatizirati takšne teste. Eno izmed takšnih ogrodji je Metasploit, ki je bil zgrajen s filozofijo, da morajo imeti penetracijski testerji na voljo enaka orodja kot kriminalci. To ogrodje je že zelo zrelo, saj se je začelo razvijati že leta 2003. Prednost Metasploit-a pred ostalimi ogrodji je predvsem v njegovi modularni arhitekturi. To ogrodje z moduli kot je Meterpreter dosega zelo visoko fleksibilnost pri uporabi le-tega ter razvoju dodatnih modulov. Z grafičnimi vmesniki kot je Armitage pa ogrodje lažje uporabljajo tisti, ki tega ogrodja ne uporabljajo vsak dan.

V seminarski nalogi sem predstavil osnove o ogrodju, napisal sem tudi nekaj o zgodovini razvoja, podrobneje sem opisal arhitekturo ogrodja, nato sem predstavil modul za izvajanje kode po izkoriščeni ranljivosti Meterpreter z njegovo prilagodljivostjo in za konec sem predstavil še grafični vmesnik Armitage. S to predstavitvijo sem želel prikazati kako pomembni so penetracijski testi za varnost in kako lahko uporabimo orodje Metasploit za izvedbo le-teh.

Pri izdelavi naloge sem se spoznal kakšen pomen ima penetracijsko testiranje za varnost in, da ni to le poganjanje avtomatskih orodij. Spoznal sem tudi kako je pri izgradnji ogrodja pomembna arhitektura, da se lahko ogrodje uporablja dlje od parih let. Poleg tega pa sem spoznal, kaj vse omogoča ogrodje Metasploit pri pomoči izvedbe penetracijskega testa in česar ne.

Za nadaljnje delo pa sem razmišljal o primerjavi takšnih in podobnih ogrodij kot je Metasploit, da se vidi, kaj pri izgradnji takšnih ogrodij deluje ter kakšne so razlike v funkcionalnostih.

Viri

A STUDY OF METASPLOIT TOOL. (2016). Pridobljeno 21. 5 2017 iz International Journal of Engineering Sciences & Research Technology: http://www.ijesrt.com/issues%20pdf%20file/Archive-2016/February-2016/41_A%20STUDY%20OF%20METASPLOIT%20TOOL.pdf

Darknet. (1. 12 2010). *Armitage – Cyber Attack Management & GUI For Metasploit*. Pridobljeno 23. 5 2017 iz Darknet: https://www.darknet.org.uk/2010/12/armitage-cyber-attack-management-gui-for-metasploit/

EternalBlue: Metasploit Module for MS17-010. (19. 5 2017). Pridobljeno 21. 5 2017 iz Rapid7: https://community.rapid7.com/community/metasploit/blog/2017/05/17/metasploit-the-power-of-the-community-and-eternalblue

Holik, F., Horalek, J., Marik, O., Neradova, S., & Zitta, S. (2014). *Effective penetration testing with Metasploit.* University of Pardubice, Faculty of electrical engineering and informatics. Budapest, Hungary: IEEE.

Maynor, D. (2011). *Metasploit Toolkit for Penetration Testing, Exploit Development, and Vulnerability Research.* Elsevier.

O'Gorman, J., Kearns, D., & Aharoni, M. (2011). *Metasploit: The Penetration Tester's Guide.* No Starch Press.

Rapid7. (26. 12 2004). *Metasploit's Meterpreter.* Pridobljeno iz Rapid7: https://dev.metasploit.com/documents/meterpreter.pdf

Rapid7. (8 2011). *Metasploit 4.0 released*. Pridobljeno 20. 5 2017 iz Rapid7: https://community.rapid7.com/community/metasploit/blog/2011/08/01/metasploit-40-released

Rapid7. (2012). *MSF User Guide 4.2*. Pridobljeno 20. 5 2017 iz http://www.nothink.org/metasploit/documentation/MSF_UserGuide_4.2.0.pdf

Rapid7. (2017). *Metasploit Editions*. Pridobljeno 20. 5 2017 iz Rapid7: https://www.rapid7.com/products/metasploit/download/editions/

Singh, A. (2012). *Metasploit Penetration Testing Cookbook.* Packt Publishing Ltd.

Skupina PTES. (2009). *PTES*. Pridobljeno iz http://www.pentest-standard.org

Wikipedia. (2017). *Metasploit Project*. Pridobljeno 20. 5 2017 iz Wikipedia: https://en.wikipedia.org/wiki/Metasploit_Project